

# 教育用システムにおける FORTRAN 用グラフィックライブラリの構築

理工学部地球環境学科 市村 雅一  
ichimu@cc.hirosaki-u.ac.jp  
理工学部地球環境学科 葛西 真寿  
kasai@phys.hirosaki-u.ac.jp

## 1 はじめに

現在総合情報処理センターの教育用システムは、Windows 環境と Linux 環境の 2 つを提供しており、Linux 環境は主に理工系の専門教育に利用されている。理工学部では、複数の学科でこの環境下での FORTRAN 言語を用いたプログラミングおよび数値解析の演習を交えた授業が行われている。FORTRAN 言語は一般的には C や Java の普及により使用者が減少しているが、初心者にも理解しやすい高級言語であることや、歴史が古く蓄積された資産が豊富であることなどを背景に一定の需要を保っている。

Linux では基本的なグラフィカル・ユーザ・インターフェースとして X Window 環境が提供されており、Xlib と呼ばれるライブラリの機能をアプリケーションから呼び出すことでユーザが自由にグラフィクス操作を行えるようになってきている。しかし、このライブラリの機能を呼び出すインターフェースは C 言語の仕様に準じているため、FORTRAN から直接呼び出すのは困難である。数値解析の演習を行う場合、計算結果をいかにわかりやすく可視化するかが問題となるが、現状では計算結果を一旦ファイルに出力してから、gnuplot など別のアプリケーションに渡してグラフにする等の処理を行っている。

そこで今回、FORTRAN 言語から Xlib の機能を呼び出すためのインターフェースとなるライブラリ (libIX0) を作成した。また、利用する上での手引きとなるドキュメントを WEB ページとして作成・公開した。このライブラリを用いれば、自作の FORTRAN プログラムから直接グラフを描画できるため、数値計算を行いながら計算結果を随時可視化できるようになる。以下、作成したライブラリの詳細、使用例について報告する。

## 2 ライブラリの概要

本ライブラリは、Intel x86 アーキテクチャの CPU を想定した上でアセンブリ言語により作成した。アセンブラとしては、総合情報処理センターの教育用システムで実際に使用されている FORTRAN コンパイラ g77 を使用した。

ライブラリ中には、機能によって関数として呼び出さなければならないもの、サブルーチンタイプのものが混在しているが、呼び出し方法や必要な引き数については後述の WEB ページに記載されている。

関数の返り値及び引き数のタイプは全て 4 バイト整数で統一している (但し文字列変数を除く)。関数名は暗黙の型宣言 (特に宣言しなければ I~N で始まる変数は整数型になるというルール) を使用しても問題が起こらないように、全て整数型の名前になっているが、引き数については使用する側が意識しておく必要がある。以下ライブラリに含まれる関数、サブルーチンの機能別一覧を示しながら説明していく。

### 2.1 ウィンドウ操作

ウィンドウの生成・表示や属性の設定などの機能を持つ部分である。図を描いたり表示したりする場所を確保するためには、最低限以下の作業が必要となる。

- 1) ディスプレイサーバとの接続 [ixopendisplay]

- 2) ウィンドウの生成 [ixcreatesimplewindow]
- 3) ウィンドウの可視化 (Mapping) [ixmapwindow]
- 4) X サーバへのリクエストの送出 [ixflush]

本ライブラリに含まれる Xlib 関連の機能を利用するためには、アプリケーションの始めの方で必ず **ixopendisplay** を使ってディスプレイサーバとの接続をしなければならない。その後、ixopendisplay の返り値として得られた Window ID を用いて図形描画等の機能を利用する。

これらの内容を理解するには、X Window に関する基本的な知識が必要とされるが、簡単な図を描くために本ライブラリを使用するだけならば、後述のサンプルプログラムを参考に、「おまじない」として使用すれば問題ない。

### ウィンドウ操作

関数・サブルーチン名	機能
ixopendisplay	ディスプレイサーバとの接続
irootwindow	Root Window ID の取得
ixcreatesimplewindow	ウィンドウの生成
ixmapwindow	ウィンドウの可視化 (Mapping)
ixmapsubwindows	サブウィンドウの可視化 (Mapping)
ixflush	リクエストの強制送出
iblackpixel	黒色ピクセル値の取得
iwhitepixel	白色ピクセル値の取得
ixquerycolor	RGB 値の取得 (XColor 構造体指定)
icpixel	ピクセル値の取得 (カラー名指定)
ixsetwindowbackground	ウィンドウの背景色指定
ixalloccolor	ピクセル値の取得 (XColor 構造体指定)
ixclearwindow	ウィンドウのクリア
ixgetgeometry	ウィンドウの位置や大きさを調べる
noredirect	ウィンドウマネージャの介入を阻止
ixsethints	ウィンドウ表示の位置、サイズの設定
ixsetinputfocus	フォーカスウィンドウの変更

## 2.2 グラフィクス操作

ウィンドウ上に図形を描画するためには、グラフィクス・コンテキスト (以下 GC と呼ぶ) が必要となる。GC は図形描画時のグラフィクス属性を保持しているリソースであり、例えば直線を描く時の線の幅、スタイル、色などの情報を GC に予めセットしておく必要がある。基本的には、以下のような手順となる。

- 1) GC の生成 [ixcreategc]
- 2) 属性の指定 [ixsetforeground など]
- 3) 図形描画 [ixdrawline など]
- 4) X サーバへのリクエストの送出 [ixflush]

## グラフィクス操作

関数・サブルーチン名	機能
ixcreategc	グラフィクス・コンテキスト (GC) の生成要求
ixfreegc	GC の開放
ixsetforeground	前景色の指定
ixsetbackground	背景色の指定
ixsetfunction	GC に対する機能の指定
ixsetlinewidth	直線の太さの指定
ixsetgraphicsexposures	GC の Exposure イベント無しの設定
ixdrawline	直線の描画
ixdrawrectangle	長方形の描画
ixfillrectangle	塗りつぶし長方形の描画
ixdrawpoint	点の描画
ixdrawarc	円弧の描画
ixfillarc	塗りつぶし円弧の描画

## 2.3 文字列描画

文字列を描画する場合の属性は、グラフィクス操作と同様に GC にセットされているものを用いる。また、特に日本語文字列 (マルチバイト文字列) を描画するためには、言語環境の設定 (OS の設定) とフォントの指定が必要である。以下に日本語文字列表示の基本的な流れを示す。

- 1) 言語環境の設定 [isetlocale]
- 2) GC の生成 [ixcreategc]
- 3) 属性の指定 [ixsetforeground など]
- 4) フォントの指定 [ixcreatefontset など]
- 5) 文字列描画 [ixmbdrawstring など]
- 6) X サーバへのリクエストの送出 [ixflush]

## 文字列描画

関数・サブルーチン名	機能
ixloadfont	フォントのロード
ixsetfont	GC へのフォントのセット
isetlocale	言語環境の設定 ( OS の関数 )
ixcreatefontset	フォントセットの作成
ixmbdrawstring	マルチバイト文字列の描画
ixmbdrawimagestring	マルチバイトイメージテキストの描画
ixdrawstring	1byte 系文字列の描画
ixdrawimagestring	1byte 系イメージテキストの描画

## 2.4 イベント関連 (キー入力, マウス関係を含む)

X Window 環境では、常にマウス、キーボード、グラフィクス等の状態を監視しており、それらに何らかの変化があるとそれを「イベント」と認識する。Xlib を利用すれば、アプリケーションからイベントの

情報を入手し、それをもとに動作を起こすことができる。本ライブラリには、イベント情報の読み込み機能を始め、キー入力やマウスの移動、ボタン押下などのイベントに関する情報入手の機能も含まれている。

#### イベント関連 (キー入力, マウス関係を含む)

関数・サブルーチン名	機能
ixselectinput	イベントの選択 (イベントマスク)
ixnextevent	イベントの読み込み (イベント発生時)
ixcheckmaskevent	イベントの読み込み (real time)
inkey	キー入力検出 (アスキーコード)
incode	キー入力検出 (キーコード)
qpointer	マウス情報の取得 (real time)
getxy	マウス情報の取得 (イベント発生時)
ixwarppointer	マウスカーソルの強制移動
ixcreatefontcursor	マウスカーソルの生成 (標準マウスカーソルを利用)
ixfreecursor	マウスカーソルの開放
ixdefinecursor	指定ウィンドウ内のマウスカーソルの定義
ixundefinecursor	マウスカーソル定義の解除
wcursor	ウィンドウ内へのマウス連動十字カーソル表示
whcur	ウィンドウ内へのマウス連動横線表示
wvcur	ウィンドウ内へのマウス連動縦線表示
wboxcur	マウス連動長方形の表示
wstarea	マウスによるウィンドウ内の領域指定

## 2.5 画像データ操作

画像データを扱う場合、その属性を保持するためのリソース (image 構造体) を生成し、それを参照しながら処理を進める必要がある。本ライブラリには、image 構造体の生成や画像処理エリアとしてのピクスマップの作成を始めとして、画像の表示、取り込み、.xwd フォーマットでの保存等の機能が含まれている。これらの機能を利用するには、X Window 上での画像データの取扱いに関する知識が必要となる。

#### 画像データ操作

関数・サブルーチン名	機能
ixgetimage	ウィンドウ中の画像の取得
ixputimage	画像データの表示
ixputimagebw	画像データの表示 (白黒 1byte 画像専用)
ixcopyarea	ウィンドウ間の画像のコピー
idefaultvisual	Default Visual の取得
idefaultdepth	ピクスマップの深さの取得
ixcreatepixmap	ピクスマップの生成
ixfreepixmap	ピクスマップの開放
ixcreateimage	image 構造体の生成
ixdestroyimage	image エリアの開放
ixcreatebitmapfromdata	ビットマップの生成
xwdsave	表示画像の .xwd file への保存

## 2.6 プロセスへのアクセス

自作した FORTRAN プログラムから、他のプロセスへのアクセスをするための機能である。この部分は Xlib とは無関係であるが、現在授業で頻繁に利用されている gnuplot を FORTRAN からコントロールすることを目的として作成した。後述の WEB ページにサンプルプログラムを含めて使用方法の解説がある。

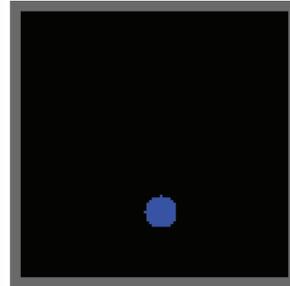
### プロセスへのアクセス

関数・サブルーチン名	機能
ipopen	プロセスの open
ipwrite	プロセスへの書き出し
ipflush	プロセス入出力バッファの flush
ipclose	プロセスの close

## 3 libIX0 の使用例

### 3.1 剛体球のアニメーション

以下に libIX0 を利用して作成したサンプルプログラムのソースリストを示す。これを実行すると、右図のような球が表示され、壁に跳ね返されながら動きまわるアニメーションが見れる。このプログラムでは、グラフィクス・コンテキストの機能を xor モードに指定し、黒の背景上で同一の図(球)を上書きすることで、図形の描画と消去を繰り返している。



```

c   剛体球の壁との弾性衝突プログラム
c   ir:球の半径  lx,ly:表示エリアの大きさ
parameter (ir=6,lx=100,ly=100)

ix=50      ! x 座標初期値
iy=50      ! y 座標初期値
ivx=-2     ! 初速度(座標増分)x成分
ivy=5      !          "          y成分
ixmin=ir   ! 中心 x 座標の最小値
ixmax=lx-ir !  "          の最大値
iymin=ir   ! 中心 y 座標の最小値
iymax=ly-ir !  "          の最大値

ichild=0
mx=0
my=0
mask=0

call ixopendisplay()      ! ディスプレイサーバとの接続
ibblue=icpixel('blue',4) ! "青" のピクセル値取得
igray=icpixel('dim gray',8) ! "灰色"          "
iwr=irootwindow()        ! Root Window ID の取得
c   ----- ウィンドウ生成 & ウィンドウ ID の取得 -----
iw=ixcreatesimplewindow(iwr,250,10,lx,ly,4,igray,ibblackpixel())

```

```

call noredirect(iw)          ! ウィンドウマネージャの介入を阻止
call ixselectinput(iw,0)    ! イベントマスクの設定
call ixmapwindow(iw)        ! ウィンドウの可視化
igc=ixcreategc(iw)          ! GCの生成
call ixsetfunction(igc,6)   ! GCの機能設定(xorモード)
call ixflush()              ! リクエスト送出

c   ----- 円を描く -----
call ixsetforeground(igc,ibblue) ! 色指定
call ixfillarc(iw,igc,ix-ir,iy-ir,2*ir,2*ir,0,360) ! 塗りつぶし円の描画
call ixflush()              ! リクエスト送出

10  continue
    call system('sleep 0.0001') ! スピード調節
c   ----- 円の消去(xorモードで同じ場所へ描画) -----
call ixfillarc(iw,igc,ix-ir,iy-ir,2*ir,2*ir,0,360)
ix=ix+ivx                    ! x座標移動
iy=iy+ivy                    ! y座標移動
call ixfillarc(iw,igc,ix-ir,iy-ir,2*ir,2*ir,0,360) ! 円描画
call ixflush()              ! リクエスト送出
if (ix.le.ixmin.or.ix.ge.ixmax) ivx=-ivx ! 壁にぶつかったらx反転
if (iy.le.iymin.or.iy.ge.iymax) ivy=-ivy ! " " y反転
call qpointer(iw,ichild,mx,my,mask) ! マウス情報取得
if (mask.ne.0) go to 999        ! ボタン押下で終了
go to 10
999 stop
end

```

---

### 3.2 重力レンズ効果

ブラックホールが土星と地球の間を横切ったら重力レンズ効果によって、どう見えるか? point mass lens モデルを用いて計算した土星の像を libIX0 を利用して視覚化した。(図1 参照)

一般相対性理論によれば、光も強い重力場によって経路が曲げられる。重力レンズ効果とは、遠方の天体からの光が手前の重力源によって進路を曲げられた結果、もともと一つの光源の像が複数個観測される現象である。図2の「レンズ効果を受けない土星の像」の各ピクセルの位置と色情報を libIX0 のサブルーチンを使って読み取り、重力レンズ方程式を解いて各ピクセルの像の位置を計算し、その位置に元画像の色情報をもったピクセルを打つという操作を繰り返す。尚、地球と土星の元画像は宇宙航空研究開発機構および国立天文台が提供している画像ファイルを利用した。

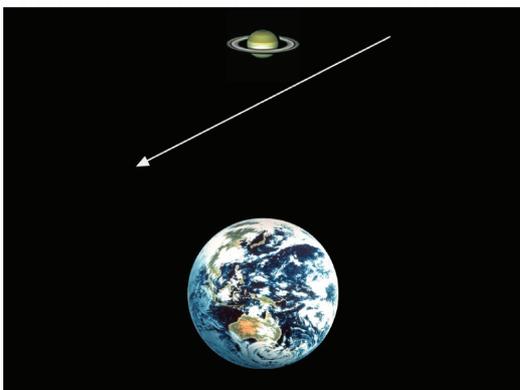


図 1: 地球と土星

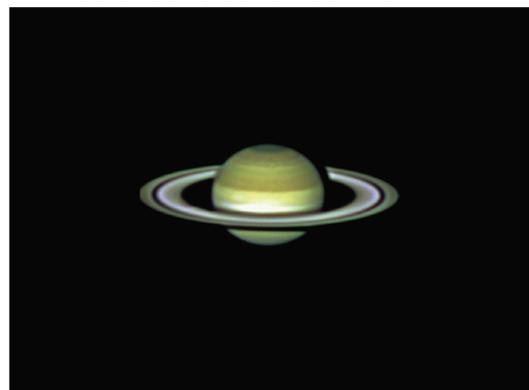


図 2: レンズ効果を受けない土星の像

上述の手法によって得られたレンズ効果を受けた像は以下のようなになる。下の図3は、土星の像が、手前を通るブラックホールの重力場によるレンズ効果によって2つに見えている様子を示している。

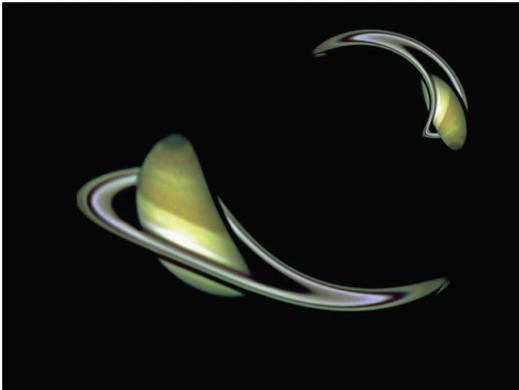


図 3: レンズ効果を受けた像

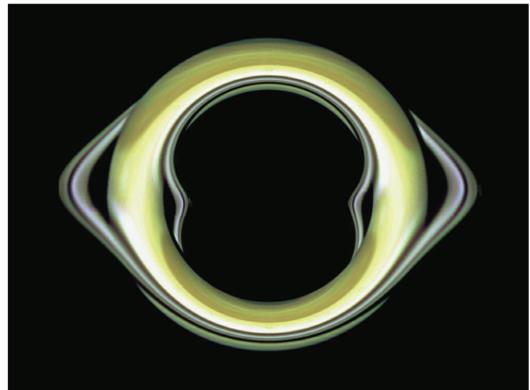


図 4: アインシュタインリング!

観測者からみて、重力レンズ天体と遠方の光源が視線方向の一直線上に並ぶ場合、重力レンズ効果によってリング状の像、アインシュタイン・リングが得られる。元々リングを持っている土星の場合はどうなるか? 上の右図のように、(土星の) リングの (アインシュタイン) リングができる!

レンズ天体となるブラックホールの位置を変えると、像の位置や形が変わる。これらの様子を java script を利用したアニメーションにまとめたものは以下の URL で見ることができる。

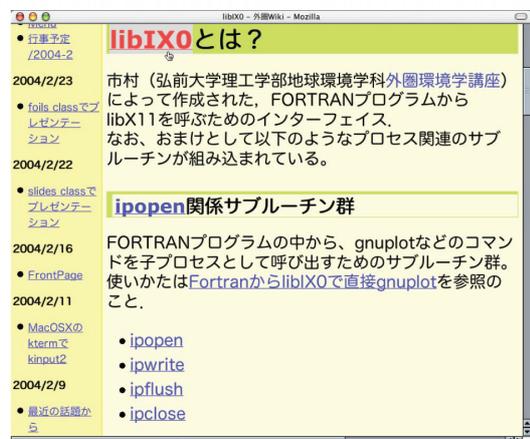
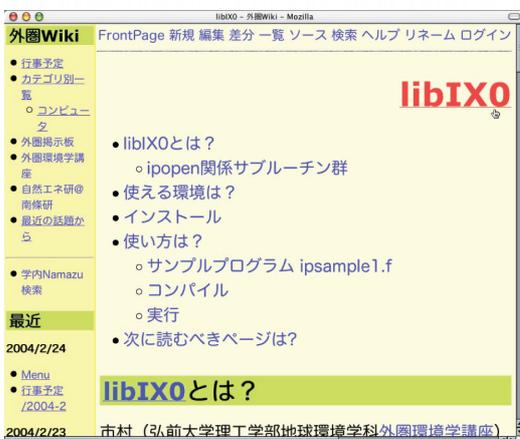
- 重力レンズでみた土星の像アニメ

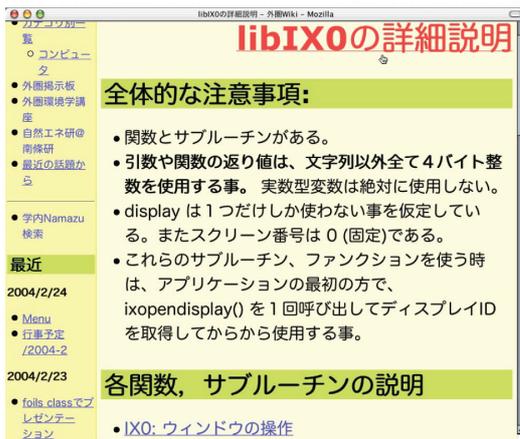
<http://windom.phys.hirosaki-u.ac.jp/member/kasai/lens-anim/>

## 4 WEB 上の libIX0 ドキュメント: 外圏 Wiki

FORTRAN 用グラフィックライブラリ libIX0 のドキュメントは、我々の研究室の WEB ページ、「外圏 Wiki」からアクセスできる。ここでは、ライブラリのインストール方法やプロセスへのアクセスを行う機能 (ipopen 関係) のサンプルプログラム等も掲載されている。

- 外圏 Wiki: <http://windom.phys.hirosaki-u.ac.jp/fswiki/>





## 5 最後に

今回作成したライブラリ libIX0 は、FORTRAN で行った数値計算の結果を、FORTRAN で可視化できるという点で非常に有用である。Xlib の機能をそのまま FORTRAN から呼び出す関数・サブルーチンが多いため、ある程度 X Window の知識が必要となるのが気になるが、これはより上位のサブルーチンを FORTRAN で作成すれば解決できる問題である。もっとも、X Window システムを学ぶための入口としては、このままで丁度良い題材となるかも知れない。

## 参考文献

- [1] 木下凌一・林秀幸 著, X-Window Ver.11 プログラミング, 日刊工業新聞社
- [2] 柴山守 著, X11 による画像処理, 技術評論社